Poly-co: a multilayer perceptron approach for coreference detection

Eric Charton
École Polytechnique de Montréal
2500, chemin de Polytechnique
Montréal (Québec), H3T 1J4
eric.charton@polymtl.ca

Michel Gagnon
École Polytechnique de Montréal
2500, chemin de Polytechnique
Montréal (Québec), H3T 1J4
michel.gagnon@polymtl.ca

Abstract

This paper presents the coreference resolution system Poly-co submitted to the closed track of the CoNLL-2011 Shared Task. Our system integrates a multilayer perceptron classifier in a pipeline approach. We describe the heuristic used to select the pairs of coreference candidates that are feeded to the network for training, and our feature selection method. The features used in our approach are based on similarity and identity measures, filtering informations, like gender and number, and other syntactic information.

1 Introduction

Coreference resolution is the process of determining whether two expressions in natural language refer to the same entity in the world. It is an important sub-task in natural language processing systems. In this paper, we present a learning approach to coreference resolution of named entities (NE), pronouns (PRP), noun phrases (NP) in unrestricted text according to the CoNLL-2011 shared task (Pradhan et al., 2011). This system have been used in the context of closed track.

2 Previous propositions

Many learning-based systems have been proposed to solve coreference resolution task, and Soon's (Soon et al., 2001) architecture is one of the most popular ones. In this proposition, all possible mentions in a training document are determined by a pipeline of natural language processing (NLP) modules. Then, training examples are generated as fea-ture vectors. Each feature vector represents a pair of mentions that can potentially corefer. Those vectors are used as training examples given to build a C5 classifier. To determine the coreference chains in a new document, all potential pairs of coreferring mentions are presented to the classifier, which decides whether the two mentions actually core-fer. Since then, this dominant architecture has been widely implemented. As it is a very flexible proposition, many families of classifiers have been used, trained with various configurations of feature vectors. Good results are obtained with SVM classifiers, like described in (Versley et al., 2008). Some propositions keep only the principle of feature vectors, associated with more complex coreference detection algorithms. A constraint-based graph partitioning system has been experimented by (Sapena et al., 2010) and a coreference detection system based on Markov logic networks (MLNs) has been proposed by (Poon and Domingos, 2008).

3 Architecture of the proposed system

A considerable engineering effort is needed to achieve the coreference resolution task. A significant part of this effort concerns feature engineering. We decided to keep the well established architecture of (Soon et al., 2001) with a pre-processing NLP pipeline used to prepare pairs of coreference features. The features are then submitted to the classifier for pairing validation. We tested various classifiers on our feature model (see table 2) and finally selected a multilayer perceptron (MLP) classifier to make decision. Since the Ontonotes layers provide syntactic information (Pradhan et al., 2007),
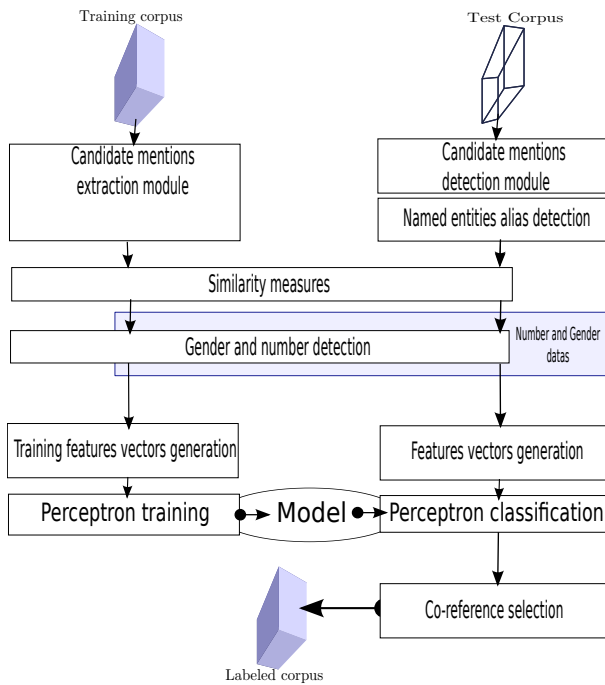
Figure 1: The pipeline architecture of the Poly-co system.

we could concentrate our efforts on the introduction of some complementary high level properties (like mention similarities or gender compatibility) used in the feature vectors given to the classifiers. The global architecture, presented in figure 1, includes two pipelines. One configured for training purposes and the other one for coreference resolution.

3.1 Architecture components

Ontonotes corpus includes part-of-speech tagging, noun phrases identification and named entity labels. We introduce complementary modules to detect gender and number, and evaluate mentions aliasing and similarity. The detection task is composed of 4 modules:

- **Candidate mentions detection module**, based on extraction rules, using Ontonotes layers.

- Named entities **alias detection module**, based on the previous version of Poly-co, described in (Charton et al., 2010). The purpose of this module is to identify variations in names of the same entity by examination of their surface form.

- **Similarity calculation module**, used to evaluate the similarity of two mentions according to

a comparison of their string.

- **Gender and number detection module**, which determines gender and number for any candidate mention.

In the training pipeline, the **candidate mentions detection module** and the **alias detection module** are replaced by a unique **candidate mentions extraction module**. This module collects from the training corpus the labeled mentions and their reference numbers and use them to generate aliases and mentions values required to build training features.

As we will see later, similarity calculation and gender and number detection all result in a value that is integrated to the feature vector used to train and apply the classifier. We give below a more detailed description of each module.

3.1.1 Candidate mentions detection module

It is mandatory for coreference resolution to first get all the potential mentions from the input text. To determine the mentions, this module explores the text corpus and extracts a *candidate mentions list*. This list includes, for each mention, its position in the document, its word content and its syntactic category. This module uses simple detection rules to collect the mentions according to their part of speech (POS) and their text content, their syntactic boundaries and their named entity type labels.

When used in classification mode, the detection process is followed by a filtering process, where rules are used to remove mentions that have a very low probability of being involved in coreference. These rules are based on simple word sequence patterns. For example, pronoun *it* is filtered out when immediately followed by verb *to be* and relative pronoun *that* within the next 6 following words.

3.1.2 Alias detection module

This module implements an algorithm that clusters entities by comparing the form of their names. Entities are put in a list, ordered according to their chronological apparition in the text. At the beginning of the process, the first entity in the list is removed and constitutes the first item of a cluster. This entity is compared sequentially, by using similarity and logical rules (i.e, a PERSON can't be an alias of a LOC), with every other entities contained in the

list. When there is a match, the entity is removed from the list and transferred to the currently instantiated cluster. This operation is repeated until the list is empty.

At the end of this process, an entity in a cluster is considered to be an alias of every other entity in the same cluster.

The TIME and DATE alias detection is done through a specific heuristic set. Each TIME entity representation is converted in a standardized format (Hour/Minutes). Dates are normalized as a relative amount of days (*"today"* is 1, *"last month"* is -30, etc) or a formal date (Year/Month/Day).

3.1.3 Similarity calculation module

The similarity module is applied on named entities (excepted TIME and DATE) and NP of the *candidate mentions list*. It consists in a text comparison function which returns the number of common words between two mentions. After execution of this module, we obtain a square matrix containing a similarity measure for every pair of mentions.

3.1.4 Gender and number detection module

Gender and number are associated with each entry of the *candidate mentions list*, including PRP and NP. First, this module tries to detect the gender using the gender data provided[1]. Then a set of less than 10 very simple rules is used to avoid anomaly (i.e a PERSON entity associated with the neutral gender). Another set of rules using plural markers of words and POS is used to validate the number.

4 Features definition and production

The feature vector of the Poly-co system (see table 1) consists of a 22 features set, described below. This vector is based on two extracted mentions, A and B, where B is the potential antecedent and A is the anaphor.

Four features are common to A and B (section *A and B properties* of table 1):

- **IsAlias** : this value is binary (yes or no) and provided by the **alias module**. The value is *yes* if A and B have been identified as describing the same entity.

---

[1]The list allowed by the Shared Task definition and available at http://www.clsp.jhu.edu/ sbergsma/Gender/

Feature Name	Value	value
A and B properties		
IsAlias	yes/no	1/0
IsSimilar	real	0.00 /1.00
Distance	int	0/const(b)
Sent	int	0/x
Reference A		
ISNE	yes/no	1/0
ISPRP	yes/no	1/0
ISNP	yes/no	1/0
NE_SEMANTIC TYPE	null / EN	0 / 1-18
PRP_NAME	null / PRP	0 / 1-30
NP_NAME	null / DT	0 / 1-15
NP_TYPE	null / TYPE	0 / 1-3
GENDER	M/F/N/U	1/2/3/0
NUMBER	S/P/U	1/2/0
Reference B		
Same as Reference A		

Table 1: Feature parameters

- **IsSimilar** : this value is the similarity measure provided by the **similarity module**.

- **Distance** : this indicates the offset distance (in terms of number of items in the *candidate mentions list*) between A and B.

- **Sent** : this indicates the amount of sentences marker (like . ! ?) separating the mentions A and B.

For each candidate A and B, a set containing nine features is added to the vector (in table 1, only properties for A are presented). First, 3 flags determine if mention is a named entity (IsNE), a personal pronoun (IsPRP) or a noun phrase (IsNP). The next six flags define the characteristics of the mention :

- NE_SEMANTIC TYPE is one of the 18 available NE types (PERSON, ORG, TIME, etc)

- PRP_NAME is a value representing 30 possible words (like *my, she, it, etc*) for a PRP.

- NP_NAME is a value indicating the DT used by a NP (like *the*, *this*, *these*, etc).

- NP_TYPE specifies if NP is demonstrative, definite, or a quantifier.

- GENDER and NUMBER flags indicate whether the mention gender (*Male, Female* or *Neutral*)

Poly-co Score	Mentions			B3			CEAF			MUC		
	R	P	F	R	P	F	R	P	F	R	P	F
Multilayer perceptron (MLP)	65.91	64.84	65.37	66.61	62.09	64.27	50.18	50.18	50.18	54.47	50.86	**52.60**
SVM	65.06	66.11	**65.58**	65.28	57.68	61.24	46.31	46.31	46.31	53.30	50.00	51.60
Tree J48	66.06	64.57	65.31	66.53	62.27	**64.33**	50.59	50.59	**50.59**	54.24	50.60	52.36

Table 2: System results obtained with scorer v4 on gold dev-set applying various classifiers on same features vectors.

Poly-co Score	Mentions			B3			CEAF			MUC		
Multilayer perceptron (MLP)	64.53	63.42	**63.97**	66.07	61.65	**63.79**	49.12	49.12	**49.12**	52.70	49.22	**50.90**

Table 3: System results obtained with scorer v4 on predicted dev-set using our system.

and number (*Singular* or *Plural*) are known or not (if not, *U* is the value for the flag).

A *null* value (0) is used when a flag doesn't have to be defined (i.e PRP flag if the mention is a NE).

5 Classifier training and use

For training, we use an algorithm that selects the more relevant pairs or mentions. Suppose that the *candidate mentions list* contains k mentions $M_1, M_2, \ldots, M_k$, in this order in the document. The algorithm starts with the last mention in the document, that is, M_k. It compares M_k sequentially with preceding mentions, going backward until a coreferring mention M_c is reached, or a maximum of n mentions have been visited (the value of n is fixed to 10 in our experiments). When a coreferring mention M_c has been found, a vector is constructed for every pair of mentions $\langle M_k, M_i \rangle$, where M_i is a mention that has been visited, including the coreferring one. These vectors are added to the training set, M_c being a positive instance, and all the others ones being negative instances. The process is repeated with M_{k-1}, and so on, until every mention has been processed. If none of the n precedent mentions are coreferent to M_1, all the n pairs are rejected and not used as training instance.

During the coreference detection process, a similar algorithm is used. Starting from mention M_k, we compare it with n preceding mentions, until we find one for which the multilayer perceptron classifier gives a coreference probability higher than 0.5[2]. If none is found within the limit of n mentions, M_k

[2]Note that in comparison tests, displayed in table 2, SVM provides a binary decision and J48 a probability value. They are used as the multilayer perceptron ones.

is considered as a non coreferring mention. When this has been done for every mention in the document, the detected coreferences are used to construct the coreference chains.

6 Results

The results presented on table 2 are obtained on the dev-set of the Ontonotes corpus. To evaluate the potential of our features model, we trained our system with MLP, SVM and J48 Tree classifiers. We finally chose the MLP models for the test evaluation due to its better performance on the predicted dev-set. However, according to the small difference between MLP and J48 Tree, it's difficult to define clearly wich one is the best choice.

7 Conclusions

We presented Poly-co, a system for coreference resolution in English easy to adapt to other languages. The first version of Poly-co was built to detect only coreferences of persons. As the dataset provided for CoNLL is much more complex, it was an intersting opportunity to evaluate our mention detection algorithms in the perspective of a full task, including difficult coreferences mentions beetween named entities, noun phrases and prepositions. Our comparison of various classifier results on dev-sets have shown that our proposition to use a multilayer perceptron as coreference chain builder can be an intersting solution, but does not introduce an important difference of performance with previously experimented classifiers.

References

Eric Charton, Michel Gagnon, and Benoit Ozell. 2010. Poly-co : an unsupervised co-reference detection system. In *INLG 2010-GREC*, Dublin. ACL SIGGEN.

Hoifung Poon and Pedro Domingos. 2008. Joint unsupervised coreference resolution with Markov logic. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing - EMNLP '08*, page 650, Morristown, NJ, USA. Association for Computational Linguistics.

Sameer Pradhan, Lance Ramshaw, Ralph Weischedel, Jessica MacBride, and Linnea Micciulla. 2007. Unrestricted coreference: Identifying entities and events in OntoNotes. In *International Conference on Semantic Computing, 2007. ICSC 2007.*, pages 446–453. IEEE.

Sameer Pradhan, Lance Ramshaw., Mitchell Marcus, Martha Palmer, Ralph Weischedel, and Xue Nianwen. 2011. CoNLL-2011 Shared Task: Modeling Unrestricted Coreference in OntoNotes. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning (CoNLL 2011)*, Portland, Oregon.

Emili Sapena, L. Padró, and Jordi Turmo. 2010. RelaxCor: A global relaxation labeling approach to coreference resolution. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, number July, pages 88–91. Association for Computational Linguistics.

Wee Meng Soon, Hwee Tou Ng, and Daniel Chung Yong Lim. 2001. A Machine Learning Approach to Coreference Resolution of Noun Phrases. *Computational Linguistics*, 27(4):521–544, December.

Yannick Versley, S.P. Ponzetto, Massimo Poesio, Vladimir Eidelman, Alan Jern, Jason Smith, Xiaofeng Yang, and Alessandro Moschitti. 2008. BART: A modular toolkit for coreference resolution. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, number 2006, pages 9–12, Marrakech. European Language Resources Association (ELRA).